

Inhaltsverzeichnis

I	Einführung	1
1	Ein erstes Beispiel	3
1.1	Projekt anlegen	3
1.2	Die erste Activity	5
1.3	Layout definieren	6
1.4	Activities aufrufen	9
1.5	Das Android-Manifest	11
1.6	Fazit	17
2	Systemaufbau	19
2.1	Architekturübersicht	19
2.2	Die Dalvik Virtual Machine	21
2.3	Standardbibliotheken	23
2.4	Der Anwendungsrahmen	24
2.5	Android-Komponenten	24
2.6	Die Klasse Context	26
3	Sicherheit	27
3.1	Das Sandbox-Prinzip	27
3.2	Signieren von Anwendungen	28
3.3	Berechtigungen	29
3.4	Anwendungsübergreifende Berechtigungen	31
II	Android in der Praxis	33
4	Beispielanwendung »Amando«	35
4.1	Aus Sicht des Anwenders	35
4.2	Kapitelübersicht	38

5	Oberflächengestaltung	39
5.1	Ziel	39
5.2	Schnelleinstieg: Activities, Layouts und Views	39
5.2.1	Grundbausteine der Oberflächengestaltung	40
5.2.2	Oberflächen implementieren	41
5.3	Ressourcen	44
5.3.1	Definition von Ressourcen	44
5.3.2	Zugriff auf Ressourcen	47
5.3.3	Textressourcen	50
5.3.4	Farbressourcen	51
5.3.5	Größendefinitionen	52
5.3.6	Bilder	52
5.3.7	Animationen	54
5.3.8	Multimediateien	55
5.3.9	Der raw-Ordner	56
5.3.10	XML-Dateien	57
5.3.11	Der Asset-Ordner	57
5.4	Layouts und Views	58
5.4.1	Definition von Layouts	58
5.4.2	Übersicht vorhandener Layouts	61
5.4.3	Übersicht vorhandener Views	65
5.4.4	Views verwenden	66
5.5	Schaltflächen und Menüs	68
5.5.1	Schaltflächen	69
5.5.2	Oberflächenereignisse	70
5.5.3	Menüs im Allgemeinen	71
5.5.4	Menüdefinition	72
5.5.5	Optionsmenüs	74
5.5.6	Kontextmenüs	75
5.5.7	Dynamische Menügestaltung	77
5.6	Formularverarbeitung	79
5.6.1	Zielsetzung	79
5.6.2	Arbeiten mit Views	80
5.7	Das Android-Manifest	84
5.8	Formatvorlagen: Styles und Themes	86
5.8.1	Styles	86
5.8.2	Themes	88

5.9	Implementierung einer Bildschirmseite	90
5.9.1	Checkliste: Bildschirmseite implementieren	90
5.9.2	Texte für Bildschirmseiten definieren	91
5.9.3	Styles und Themes definieren	93
5.9.4	Weitere Ressourcen definieren	93
5.9.5	Layouts definieren	93
5.9.6	Menüs definieren	95
5.9.7	Activity implementieren	95
5.9.8	Android-Manifest anpassen	97
5.9.9	Bildschirmseite im Emulator testen	98
5.10	Tipps und Tricks	100
5.10.1	Scrolling	100
5.10.2	Umgebungsabhängige Ressourcen	102
5.10.3	Hilfeseiten mit WebView darstellen	104
5.10.4	Der Hierarchy Viewer	108
5.11	Fazit	110
6	Oberflächen und Daten	111
6.1	Zielsetzung	111
6.2	AdapterViews und Ressourcen	112
6.3	AdapterViews und Adapter	113
6.3.1	ArrayAdapter	114
6.3.2	SimpleCursorAdapter	116
6.3.3	Auf Ereignisse reagieren	117
6.4	Performante Listen	120
6.5	Anwendungseinstellungen	123
6.5.1	Begriffsdefinitionen	124
6.5.2	Einstellungen definieren	124
6.5.3	Einstellungsseite implementieren	127
6.5.4	Auf Einstellungen zugreifen	129
6.5.5	Einstellungen bearbeiten	130
6.6	Fortschrittsanzeige	131
6.7	Fazit	132
7	Intents	135
7.1	Warum gibt es Intents?	135
7.2	Explizite Intents	136
7.3	Implizite Intents	137
7.3.1	Intent-Filter für implizite Intents	137
7.3.2	Ein einfaches Beispiel	140
7.3.3	Intent-Resolution	142
7.3.4	Beispiele für implizite Intents	142
7.3.5	Fehlerbehandlung	149

7.4	Sub-Activities	150
7.4.1	Sub-Activities aufrufen	150
7.4.2	Sub-Activities verwenden	151
7.5	Anruf einleiten	155
7.6	Fazit	156
8	Hintergrundoperationen	159
8.1	Ziel	159
8.2	Theorie: Prozesse, Threads, Services	160
8.2.1	Allgemeines	160
8.2.2	UI-Thread	160
8.2.3	ANR	161
8.2.4	Prozesse vs. Threads	162
8.2.5	Services	163
8.2.6	Services vs. Threads	164
8.2.7	Binder	165
8.3	Implementierung eines Local Service	168
8.3.1	Local Service	168
8.3.2	Mit einem Service verbinden	169
8.3.3	Services starten und stoppen	172
8.4	Tipps und Tricks	173
8.5	Callback-Mechanismen	174
8.5.1	Handler	175
8.5.2	Implementierung	177
8.5.3	Threads mit Schleifen	182
8.6	Fazit	184
9	IPC – Inter Process Communication	187
9.1	AIDL	187
9.2	Implementierung	188
9.3	Eigene Datenobjekte per IPC übertragen	192
9.3.1	Eigene Datenobjekte erstellen	192
9.3.2	Parameterübergabe optimieren	195
9.4	Asynchrone Methodenaufrufe	196
9.4.1	Asynchrone Methodenaufrufe mit Rückgabewert	197
9.4.2	Asynchrone Methodenaufrufe verwenden	201
9.5	Fazit	204
10	Systemnachrichten	205
10.1	Broadcast Intents	205
10.2	Broadcast Receiver	206
10.2.1	Dynamische Broadcast Receiver	208
10.2.2	Statische Broadcast Receiver	210

10.3	Meldungen an den Notification Manager	220
10.4	Fazit	224
11	Datenbanken	225
11.1	Zielsetzung	225
11.2	Android: Wozu Datenbanken?	225
11.3	Das Datenbanksystem SQLite	226
11.4	Eine Datenbank erstellen	227
	11.4.1 Berechtigungen	227
	11.4.2 Schemaverwaltung	227
11.5	Datenzugriff programmieren	230
	11.5.1 SQLiteDatabase – Verbindung zur Datenbank	231
	11.5.2 Datenbankabfragen	232
	11.5.3 Ergebnistyp Cursor	237
	11.5.4 Änderungsoperationen	239
11.6	Datenzugriff per Kommandozeile	242
11.7	Implementierung	244
	11.7.1 Ein Architekturvorschlag	244
	11.7.2 Das Schema erstellen	247
	11.7.3 Anfrageergebnisse an der Oberfläche darstellen	248
11.8	Spielzeit!	250
12	Dateisystem	251
12.1	Aufbau des Dateisystems	251
	12.1.1 Das Anwendungsverzeichnis	251
	12.1.2 SD-Karten	251
12.2	Verwaltung	252
12.3	Programmierung	253
	12.3.1 Zugriff auf das Anwendungsverzeichnis	254
	12.3.2 Zugriff auf die SD-Karte	257
12.4	Zusammenfassung	258
13	Content Provider	259
13.1	Zielsetzung	259
13.2	Übersicht/Grundbegriffe	259
13.3	Content-URIs	261
13.4	Content Provider	263
	13.4.1 Stammdaten	264
	13.4.2 Operationen	264
	13.4.3 Lebenszyklus	266
	13.4.4 Berechtigungen	266
	13.4.5 Deployment	266

13.5	Content Consumer	267
13.5.1	Content Resolver.....	268
13.5.2	Zugriff auf Datenbankinhalte	269
13.5.3	Zugriff auf Dateien	270
13.6	Implementierung	271
13.6.1	Zugriff auf das Android-Adressbuch	271
13.6.2	Der FotoProvider.....	273
13.6.3	FotoSpeicher als Content Consumer	280
13.7	Content Provider für Fortgeschrittene	282
13.7.1	Asynchrone Operationen	282
13.7.2	Alternativen zum Content Provider	284
14	Lebenszyklen	285
14.1	Prozessverwaltung	286
14.2	Lebenszyklus einer Activity	287
14.3	Lebenszyklus eines Service	290
14.4	Lebenszyklus eines Broadcast Receivers	291
14.5	Activities: Unterbrechungen und Ereignisse	292
14.6	onPause() vs. onSaveInstanceState(Bundle outState)	293
14.7	Beispiele aus der Praxis	298
14.7.1	Beispiel: Kalender-Activity	298
14.7.2	Beispiel: E-Mail-Programm	300
14.7.3	Beispiel: Quick-and-dirty-Alternative	301
15	Datenübertragung	303
15.1	Ziel	303
15.2	Theoretische Grundlagen	304
15.2.1	Das Emulator-Netzwerk	304
15.2.2	Die Internet-Einbahnstraße	306
15.2.3	Netzwerkunterstützung bei Android	307
15.2.4	Arten der Netzwerkübertragung.....	309
15.3	Netzwerken in der Praxis.....	309
15.3.1	Verfahren 1: Geoposition senden	310
15.3.2	Verfahren 2: dauerhafte Verbindung	311
15.3.3	Auf Funklöcher reagieren	315
15.4	Fazit	317
16	Standortbezogene Dienste	319
16.1	Ziel	319
16.2	Theoretische Grundlagen	320
16.2.1	GPS, KML und GPX	320
16.2.2	Entwickeln im Emulator	321
16.2.3	Debug Maps API-Key erstellen.....	322

16.3	Praxisteil	324
16.3.1	Vorbereitung	324
16.3.2	Der Location Manager	325
16.3.3	Google Maps	327
16.3.4	MapActivity	329
16.4	Fazit	336

III Android für Fortgeschrittene 339

17	Debugging und DDMS	341
17.1	Anschluss eines Android-Geräts	341
17.2	Systemausgaben mit der LogCat	342
17.3	DDMS: Dalvik Debug Monitor Service	345
17.3.1	Emulator Control	346
17.3.2	Debugging	347
17.4	Traceview	347
18	Sicherheit und Verschlüsselung	351
18.1	Motivation	351
18.2	Grundbegriffe der Verschlüsselung	352
18.2.1	Verschlüsselte Datenübertragung	353
18.2.2	Daten oder Objekte verschlüsseln	368
18.2.3	Verschlüsselung anwenden	370
19	Automatisiertes Testen	373
19.1	Was testen wir?	373
19.2	Oberflächentests	375
19.2.1	Instrumentierung und robotium	376
19.2.2	Ein Beispiel	378
19.2.3	Konflikte vermeiden	379
19.3	Modultests	380
19.3.1	Androidfreie Klassen	380
19.3.2	Androidabhängige Klassen	381
19.3.3	Ein Beispiel	382
19.4	Eine Teststrategie	385
20	Anwendungen marktreif machen	387
20.1	Hintergrundwissen	387
20.2	Das Eclipse-Plugin verwenden	388
20.3	Anwendungen mittels Kommandozeile marktreif machen ...	389
20.3.1	Ein eigenes Zertifikat erstellen	389
20.3.2	Eine Android-Anwendung signieren	391

21	Optimierung und Performance	393
21.1	Erste Optimierungsregeln	393
21.2	Datenobjekte	394
21.3	Cursor oder Liste?	394
21.4	Time is Akku!	395
Anhang	397
Literaturverzeichnis	401